

Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Currently Amended) A method for responding to a request, comprising:

accepting the request;

generating the control tree from a factory based on the request;

mapping the request to a control tree wherein the control tree is a logical representation of a graphical user interface (GUI) and wherein the control tree includes a set of controls which are related hierarchically to one another including at least one portlet control that represents at least one portlet;

advancing the control tree through at least one lifecycle stage based on the request, wherein the set of controls in the control tree includes at least one portlet control that represents at least one portlet operates to interact with each other and produce response based on the request in the at least one lifecycle stage;

providing the request to a portlet container that contains the at least one portlet; and

aggregating the output of each of the at least one portlets and providing the output to the GUI.

2. (Canceled).

3. (Original) The method of claim 1, further comprising:

generating a response wherein the response can be used to render at least a portion of the GUI.

4. (Original) The method of claim 2 wherein the step of generating a control tree from the factory comprises:

creating a metadata representation of a control tree; and

generating a class to construct the control tree based on the metadata representation.

5. (Original) The method of claim 1 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

6. (Original) The method of claim 3, further comprising:

providing the response to a web browser.

7. (Original) The method of claim 1 wherein:

the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.

8. (Original) The method of claim 1 wherein:

each one of the set of controls can have an interchangeable persistence mechanism.

9. (Original) The method of claim 1 wherein:

each one of the set of controls can render itself according to a theme.

10. (Original) The method of claim 1 wherein:

each one of the set of controls can interact with another one of the set of controls.

11. (Original) The method of claim 1 wherein:

one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.

12. (Original) The method of claim 1 wherein:

a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

13. (Original) The method of claim 3 wherein:

the response is an hypertext transfer protocol (HTTP) response.

14. (Original) The method of claim 1 wherein:

controls can raise events and respond to events.

15. (Original) The method of claim 1 wherein:

each one of the set of controls can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

16. (Currently Amended) A method for responding to a request, comprising:

accepting the request;

generating the control tree from a factory based on the request;

mapping the request to a control tree wherein the control tree is a logical representation of a portal graphical user interface (GUI) and wherein the control tree includes a set of controls which are related hierarchically to one another including at least one portlet control that represents at least one portlet;

advancing the control tree through at least one lifecycle stage based on the request, wherein the set of controls in the control tree includes at least one portlet control that represents at least one portlet operates to interact with each other and produce response based on the request in the at least one lifecycle stage;

providing the request to a portlet container that contains the at least one portlet;

aggregating the output of each of the at least one portlets and providing the output to the portal GUI; and

generating a response wherein the response can be used to render at least a portion of the GUI.

17. (Canceled).

18. (Original) The method of claim 16 wherein the step of generating a control tree from the factory comprises:

creating a metadata representation of a control tree; and

generating a class to construct the control tree based on the metadata representation.

19. (Original) The method of claim 16 wherein:
- the request is a hypertext transfer protocol request (HTTP); and
 - the request originates from a web browser.
20. (Original) The method of claim 16, further comprising:
- providing the response to a web browser.
21. (Original) The method of claim 16 wherein:
- the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.
22. (Original) The method of claim 16 wherein:
- each one of the set of controls can have an interchangeable persistence mechanism.
23. (Original) The method of claim 16 wherein:
- each one of the set of controls can render itself according to a theme.
24. (Original) The method of claim 16 wherein:
- each one of the set of controls can interact with another one of the set of controls.
25. (Original) The method of claim 16 wherein:
- one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.

26. (Original) The method of claim 16 wherein:

a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

27. (Original) The method of claim 16 wherein:

the response is an hypertext transfer protocol (HTTP) response.

28. (Original) The method of claim 16 wherein:

controls can raise events and respond to events.

29. (Original) The method of claim 16 wherein:

each one of the set of controls can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

30. (Currently Amended) A machine readable storage medium having instructions stored

thereon that when executed by a processor cause a system to:

accept the request;

generating the control tree from a factory based on the request;

map the request to a control tree wherein the control tree is a logical representation of a graphical user interface (GUI) and wherein the control tree includes a set of controls which are

related hierarchically to one another including at least one portlet control that represents at least one portlet;

advance the control tree through at least one lifecycle stage based on the request, wherein the set of controls in the control tree ~~includes at least one portlet control that represents at least one portlet~~ operates to interact with each other and produce response based on the request in the at least one lifecycle stage;

provide the request to a portlet container that contains the at least one portlet; and
aggregate the output of each of the at least one portlets and providing the output.

31. (Canceled).

32. (Original) The machine readable medium of claim 30, further comprising instructions that when executed cause the system to:

generate a response wherein the response can be used to render at least a portion of the GUI.

33. (Original) The machine readable medium of claim 30, further comprising instructions that when executed cause the system to:

create a metadata representation of a control tree; and
generate a class to construct the control tree based on the metadata representation.

34. (Original) The machine readable medium of claim 30 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

35. (Original) The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:

provide the response to a web browser.

36. (Original) The machine readable medium of claim 30 wherein:

the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.

37. (Original) The machine readable medium of claim 30 wherein:

each one of the set of controls can have an interchangeable persistence mechanism.

38. (Original) The machine readable medium of claim 30 wherein:

each one of the set of controls can render itself according to a theme.

39. (Original) The machine readable medium of claim 30 wherein:

each one of the set of controls can interact with another one of the set of controls.

40. (Original) The machine readable medium of claim 30 wherein:

one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.

41. (Original) The machine readable medium of claim 30 wherein:

a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

42. (Original) The machine readable medium of claim 32 wherein:

the response is an hypertext transfer protocol (HTTP) response.

43. (Original) The machine readable medium of claim 30 wherein:

controls can raise events and respond to events.

44. (Original) The machine readable medium of claim 30 wherein:

each one of the set of controls can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

45. (Canceled).

46. (New) A method for rendering a graphical user interface (GUI), comprising:

accepting a request;

mapping request to a control tree factory;

generating a control tree from the factory;

evaluating the control tree based on the request; and

providing a response,

wherein the control tree represent a particular instance of a control taxonomy and a control within the control tree operates to process the request, interact with each other and produce a response.

47. (New) The method of claim 46 further comprising:

running the control tree through a sequence of one or more lifecycles by employing an interchangeable lifecycle driver.

48. (New) The method of claim 46 wherein:

a “wire-up” service is used in the control tree factory that cause the control tree factory to return a root of a control tree.

49. (New) The method of claim 46, further comprising:

associating a context with a root of the control tree.